**TECHNICAL UNIVERSITY OF MOLDOVA**

REPORT

Lab. 1 – OS Simulator

**DISCIPLINE:** SOMIPP

**Student :**

**Novitchii**

**Nichita**

**Group :** FAF-193

Chișinău 2021

[Introduction 3](#_bookmark0)

[Task 4](#_bookmark1)

[Implentation 4](#_bookmark2)

[Code 5](#_bookmark3)

[Create file 5](#_bookmark4)

[Delete file 5](#_bookmark5)

[Display tree 5](#_bookmark6)

[System works 6](#_bookmark7)

[Conclusion 7](#_bookmark8)

# Introduction

A **command-line interface** (**CLI**) processes [commands](https://en.wikipedia.org/wiki/Command_(computing)) to a computer program in the form of lines of text. The program which handles the interface is called a **command-line**

**interpreter** or **command-line processor**. Operating systems implement a command-line interface in a [shell](https://en.wikipedia.org/wiki/Shell_(computing)) for interactive access to operating system functions or services.

Operating system (OS) command-line interfaces are usually distinct programs supplied with the operating system. A program that implements such a text interface is often called a command-line interpreter, command processor or [shell.](https://en.wikipedia.org/wiki/Shell_(computing))

A CLI is used whenever a large vocabulary of commands or queries, coupled with a wide (or arbitrary) range of options, can be entered more rapidly as text than with a pure GUI. This is typically the case with [operating system command shells](https://en.wikipedia.org/wiki/Operating_system_shell) CLIs are also used by systems with insufficient resources to support a graphical user interface. Some computer language systems (such as [Python,](https://en.wikipedia.org/wiki/Python_(programming_language)) [Forth](https://en.wikipedia.org/wiki/Forth_(programming_language)), [LISP](https://en.wikipedia.org/wiki/LISP), [Rexx,](https://en.wikipedia.org/wiki/Rexx) and many dialects of [BASIC](https://en.wikipedia.org/wiki/BASIC)) provide an interactive command- line mode to allow for rapid evaluation of code.

CLIs are often used by programmers and system administrators, in engineering and scientific environments, and by technically advanced personal computer users.

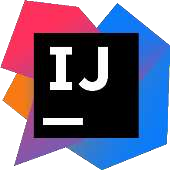
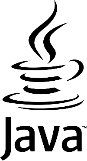
# Task

### *"OS Simulator"*

*Create an simple application (in any programming language) which will simulate an simple Command Line Operating System. It should include short "booting part of the PC or of the OS and after that it should include "simulation" of an CLI OS with a few text commands (3-6 different commands).*

*This should be your individual VISION of an simple CLI OS!*

# Implentation



I’ve decided to use java for this laboratory work, so my vision of this CLI simulator is a simple java

CLI app. So my OS have a short 3 second boot(simulation), and it has also 5 commands .

**create name** - creates a file

**delete name** - deletes a file

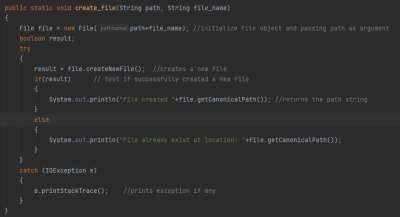
**path name [default]** - sets a path on your machine | [default option sets the default path]

**display [name]** - displays the tree of the current path | [name option used to view another path directory tree]

**exit** - exits system

# Code

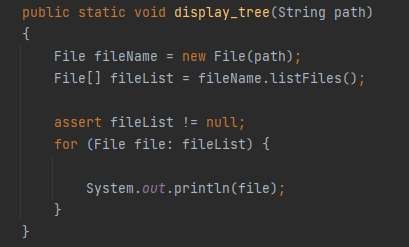
## Create file



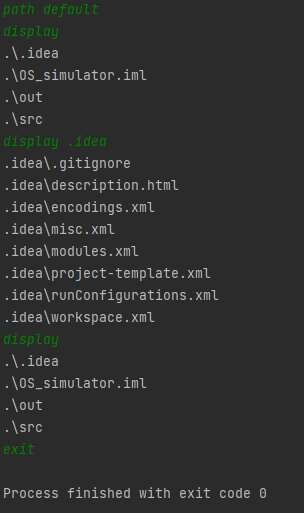
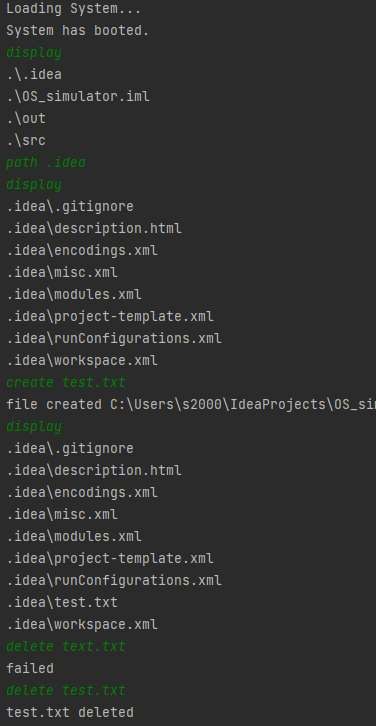
## Delete file



## Display tree



# System works



# Conclusion

This laboratory work helped us to develop simple cli apps , and understand how are organized such systems.

As for me that’s a big work to develop a full cli with instructions, and that’s really interesting for us as developers to know it’s structure and logic.